

Mémoire

”Analyse de site Web”

Réalisé par :

*Jeremy HOLAY, Lydia KAHOUADJI, Anass
EL-GHARBAOUI, Augustine CARPENTIER & Dihia
AZZEDINE*

Projet encadré par :

M. Bruno MACADRE

Remerciements

Nous souhaitons remercier l'ensemble de l'équipe pédagogique de la 3^è année de Licence Informatique de l'Université de Rouen pour leurs cours et l'apprentissage qu'ils nous ont fournis. Nous remercions plus spécialement M. MACADRE qui nous a soumis le projet annuel « Analyse Web » et qui a su nous fournir de l'aide lorsque nous en avons demandé, ainsi que Mme SELMI qui a organisé et supervisé le déroulement de l'ensemble des projets de notre promotion.

Résumé

Le projet "Analyse Web" consiste en la refonte d'une application permettant l'analyse de sites Web et l'extraction de statistiques techniques et fonctionnelles liées à eux. Cette modification passe par une amélioration de l'analyse, en intégrant le système de virtualisation léger que sont les conteneurs. Chaque conteneur doit exécuter l'analyse d'une et une seule page Web, déléguant à un nouveau conteneur chaque analyse d'une nouvelle page liée à l'actuelle. L'objectif est de réduire le temps d'exécution du programme. L'application est construite à l'aide du framework PHP Laravel et fonctionne sur un serveur Web possédant une base de donnée MySQL. Bien que le projet ne soit pas terminé, nous avons procédé avec méthode pour avancer malgré les nombreuses difficultés techniques rencontrées. Mots-clés : Site Web, Virtualisation, Optimisation, Laravel,

Docker The project "Analyse Web" is the revision of an existing application allowing the analyze of web sites and the retrieval of their technical and functional statistics. This revision upgrade is an improvement of the analyze, by adding a system of lite virtualisation named containers. Each container haves to execute the analyze of one and only one web page, delegating to a new container each new analyse bounded to the current one. The goal is to minimize the running time of the program. The application is built using the PHP framework Laravel and work on a web Server linked to a MySQL database. Even though the project is not finished, we proceed with logic to go forward and overcome the technical difficulties.

Keywords : Web Sites, Virtualisation, Optimisation, Laravel, Docker

Table des matières

Introduction	6
Présentation Générale du Projet	7
Présentation de l'existant	7
Organisation du travail	7
GIT	7
GANTT	7
L'environnement	8
Apache	8
Wamp	9
Xampp	9
Composer	9
La mise en oeuvre de l'analyseur web	10
PHP	10
Serveur MySql	10
Le rôle de MySql	10
MySql et php	10
Laravel	11
Organisation de l'application Laravel	11
Installation	11
Le routage	12
Programmation Orientée Objet	12
L'architecture Model-Vu-Contrôleur	12
Extraction du moteur d'analyse	13
Le Docker	14
Installation	14
Implémentation	15
-Implémentation du fichier docker-compose.yml	15
-Implémentation et la configuration Nginx	16
-La mise en place du Dockerfile	16
-La construction	16
Les avantages	17
La différence entre Le Docker et La VM	17
La virtualisation	18
Conclusion	19

Introduction

Dans le cadre de la 3^è année de Licence Informatique à l'Université de Rouen, notre groupe est amené à effectuer un projet nommé « Analyse Web » proposé par M. MACADRE. Ce projet consiste en l'amélioration d'une application existante, conçue au cours d'une année précédente. Le produit initial est un analyseur qui charge et scan toutes les pages d'un site web de façon séquentielle afin d'en extraire certaines statistiques telles que le temps de chargement ou encore le nombre de liens morts. Cependant, à l'exécution, plus le site possède un grand nombre de pages, plus l'analyse mettra du temps à s'effectuer. Notre mission est donc de reprendre l'application existante et de l'améliorer afin d'effectuer un traitement parallèle des pages du site web. L'idée est de réduire le temps d'analyse. Il nous a été proposé pour cela d'isoler le moteur d'analyse et de le virtualiser dans un conteneur à l'aide du logiciel Docker. Ainsi, lorsque l'application se lancera, la première page du site est analysée dans un système virtualisé, et chaque nouvelle page à analyser entraîne la création d'un nouveau conteneur à qui la tâche est déléguée. Lorsque le traitement d'une page est fini, le système renvoi les résultats spécifiques à celui-ci et ses ressources sont libérées. A la fin, l'ensemble des résultats sont rassemblés et mis à disposition de l'utilisateur. Dans ce contexte, les conteneurs sont un outil nous servant à séparer chaque analyse en une multitude d'autres, et à laisser le système d'exploitation de l'ordinateur gérer le multi-tâche lui-même, d'une façon bien plus efficace que ce que nous pourrions produire. Nous commencerons par présenter la mise en place du projet existant ainsi que son fonctionnement, notamment l'utilisation du framework Laravel et de l'architecture Modèle-Vue-Contrôleur. Nous poursuivrons en détaillant ce qu'est la virtualisation utilisée et son intégration dans le projet à l'aide de Docker, avant de conclure sur le travail réalisé.

Présentation générale du projet

Présentation de l'existant

Le projet intitulé « Outil Analyse De Site Web » a été réalisé par monsieur Sofiane Meziani et monsieur Rani Totonji en 2019 en guise de projet de fin de cycle. Le projet a été développé avec le frameworks Laravel qui est un framework PHP et Bootstrap qui est un framework CSS en utilisant les langages de programmations suivants : PHP, HTML, CSS , JavaScript. Le travail effectué a comme but de mettre en place une application en ligne performante pour analyser un site web en ayant son adresse, l'application se divise en deux différentes parties pour bien mener l'analyse : le coté Web :

- Lancer une analyse pour un site.
- Gérer les utilisateurs (inscription, connexion).
- Gérer les sites à analyser et les résultats.
- Possibilité de gérer des options d'analyse par site.

Le coté analyse :

- analyser les Statistiques sur le site (nombre de pages ...)
- Analyse performante du site (erreur de balises...)
- fournir le résultat de l'analyse sous forme de tableau
- sortir des listings
- recevoir le rapport complet de l'analyse par mail.

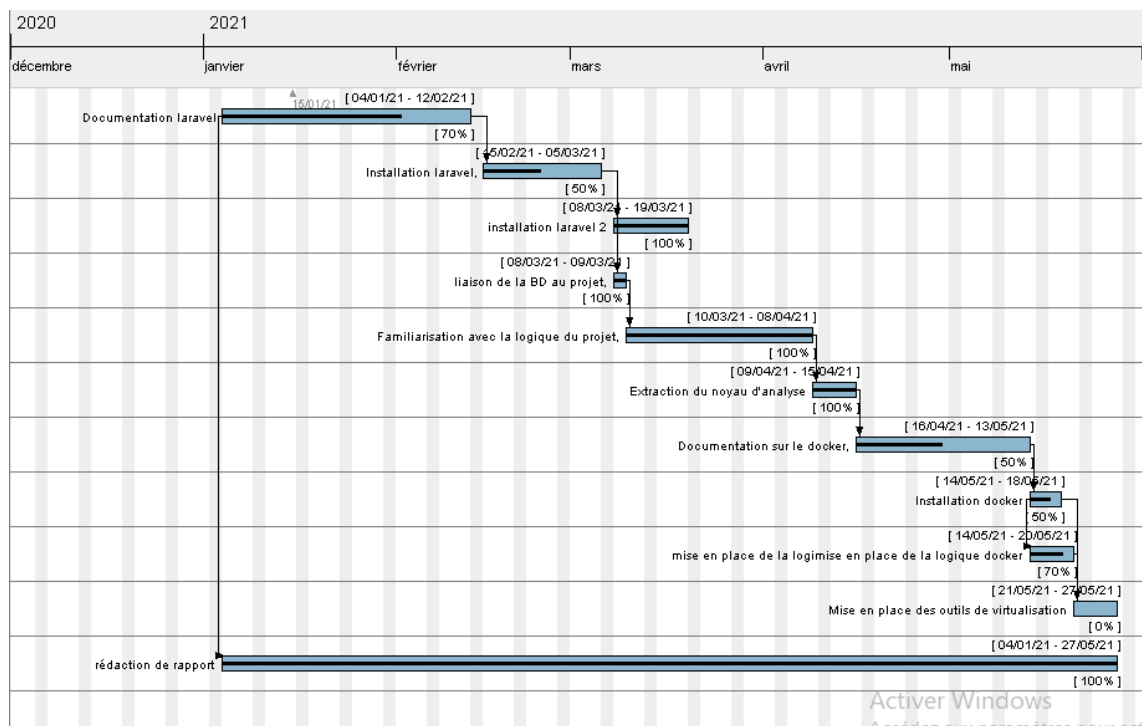
Organisation du travail

GIT

est un logiciel open-source de gestion de version décentralisée, il utilise un système de connexion pair à pair (modèle d'échange en réseau où chaque entité est à la fois client et serveur). Le code informatique développé est stocké sur l'ordinateur de chaque contributeur du projet ou sur un serveur dédié. C'est un outil de bas niveau qui se veut simple et performant, dont la principale tâche est de gérer l'évolution du contenu d'une arborescence.

GANTT

Voici un diagramme utilisé principalement pour la gestion de projet, il représente visuellement l'avancement chronologique du projet ainsi que les tâches réalisées.



L'environnement

Pour la mise en oeuvre de ce projet, on a travaillé sur deux systèmes d'exploitation différents ; UNIX et Windows. Pour assurer le fonctionnement de l'application sur l'hôte local, on a utilisé des différents environnements de développements, XAMPP et WAMP pour les différentes machines Windows et APACHE pour les machines linux.

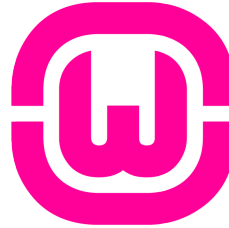
Apache

Apache est un projet de source libre, il s'agit d'un serveur web http pour les systèmes d'exploitation modernes comme UNIX et Windows. Le but de ce projet et de fournir un serveur extensible, efficace et sécurisé qui assure les services http compatibles avec la standardisation http actuelle.



Wamp

WampServer est un environnement de développement Web sous Windows pour des applications Web basé sur serveur Apache2, de langage PHP et de base de données MySQL. Ainsi Il possède la plate-forme PHPMyAdmin pour faciliter la gestion des bases de données.



Xampp

Xampp est un environnement de développement PHP de source libre le plus connu, il s'agit d'un serveur multi-plateforme se constitue principalement d'un serveur http apache, et de base de données MariaDB, Il possède la plate-forme PHPMyAdmin.



Composer

Le composer est un outil de gestion des dépendances en PHP, il permet la déclaration et la gestion des bibliothèques et des dépendances du projet.



La mise en oeuvre de l'analyseur web

PHP

Est un langage de programmation open-source, utiliser pour produire des pages web via un serveur HTTP ainsi que le traitement d'information en provenance de formulaire, il est également connu pour dialoguer avec une base de données comme MYSQL et il peut intégré facilement au HTML.



Serveur MySql

Est un serveur de gestion de bases de données relationnelles qui fonctionne sur plusieurs systèmes d'exploitation différents incluant Windows, Linux, Mac OS X, il fait parti des logiciels de bases de données les plus utilisés au monde autant par le grand public que par des professionnels.



Le rôle de MySql

Son rôle est de stocker les données, sous forme de tables, et de permettre la manipulation de ces données à travers le langage de requête SQL. Dans notre projet notre base de donnée s'appelle 'analyseweb' qui contient trois tables que nous avons créé grâce à la requete "CREATE TABLE". Le remplissage de ces tables peut se faire en français mais les requetes restent en anglais.

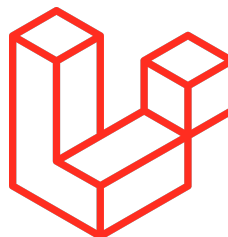
MySQL et PHP

php est capable de passer à MySQL de telles requêtes à travers les fonctions de préfixe mysql (PHP possède également des fonctions pour dialoguer avec d'autres systèmes de gestion de base

de données). C'est une raison du succès du couple PHP+MySQL dans la mise en place de sites webs.

Laravel

est un framework php qui permet de créer des applications rapidement, il respecte le principe model-vu controleur développé en programmation orienté objet. Ses ressources hébergés sur github, son installation est basée sur le gestionnaire de paquet composer.



Organisation de l'application Laravel

Laravel est organisé en plusieurs dossiers dont :

1. Le dossier App :
 - Console : toutes les commandes en mode console
 - exceptions : pour gérer les erreurs d'exécution,
 - Http : tout ce qui concerne la communication comme les controleurs
 - Providers : tous les fournisseurs de services
 - Models : le dossier des modèles avec déjà un présent
 - qui concerne les utilisateurs.
2. Les autres dossiers :
 - bootstrap :initialisation de Laravel pour le chargement automatique des classes, la fixation de l'environnement et des chemins, et pour le démarrage de l'application,
 - config :toutes les configurations : application, authentification, cache, base de données...
 - public : Le visuel du dossier public du site : CSS, images...
 - databases : migrations et populations
 - resources :vues, fichiers de langage et assets
 - routes : la gestion des urls d'entrée de l'application
 - storage : données temporaires de l'application : vues compilées, caches, clés de sessions
 - tests : fichiers de tests unitaires
 - vendor : tous les composants de Laravel et de ses dépendances (créé par composer).
3. Les fichiers de la racine :
 - artisan : outil en ligne de Laravel pour des tâches de gestion
 - composer.json :fichier de référence de composer
 - package.json : fichier de référence de npm pour les assets
 - phpunit.xml : fichier de configuration de phpunit (pour les tests unitaires)
 - .env : fichier pour spécifier l'environnement d'exécution .

Installation

Il existe plusieurs manières pour installer Laravel, on va présenter l'installation et la création d'une application Laravel via un composeur ce qu'on va utiliser ultérieurement pour lancer l'application. Pour installer et créer un projet Laravel, on suit les étapes suivante :

- On installe le composeur à partir de ce lien : <https://getcomposer.org/download/>.

- On crée un projet Laravel par utiliser le composer directement par l'exécution de la commande suivante depuis un terminal pointer au dossier racine de l'application : `composer create-project laravel/laravel myApp`.
- Après la création de l'application on peut la lancer localement à partir de la commande d'Artisan serve : `cd myApp php artisan serve`.

Le routage

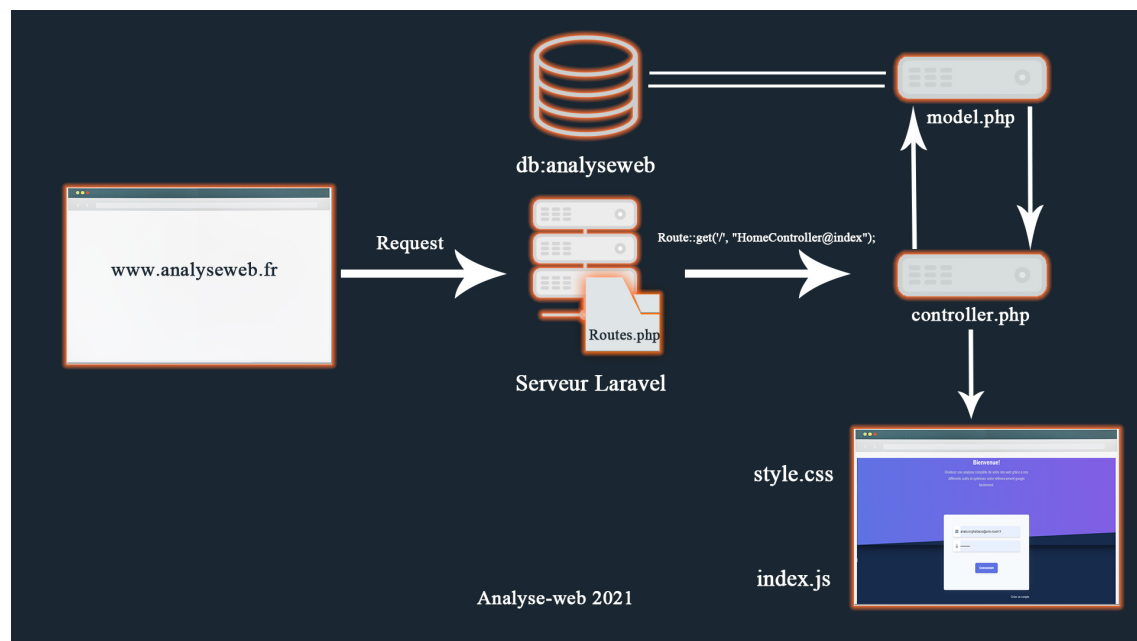
est l'un des composants clés du framework Laravel, c'est un mécanisme qui affectue le mappage de requetes vers une action de controleur spécifique. Le routage crée l'URL de l'application, toutes les routes Laravel sont définies dans le fichier « routes » automatiquement chargé par le framework. Le dossier « route » contient 4 fichier `api.php`(utiliser pour gérer les routes de l'API), `channels.php`, `console.php` et `web.php`(gère les routes normales). Pour créer une route on appelle la classe Route avec la methode HTTP souhaitée puis indiquer à cette méthode l'URL concerné et le retour à afficher pour le visiteur.

Programmation Orientée Objet

Comme on l'a précisé précédemment Laravel est orienté objet, ce dernier consiste en la définition et l'interaction de briques logicielles appelées objet ; un objet représente un concept, une idée ou une entité physique Il possède une structure interne et un comportement, et il sait interagir avec ses pairs. Il s'agit donc de représenter ces objets et leurs relations ; l'interaction entre les objets via leurs relations permet de concevoir et réaliser les fonctionnalités attendues, de mieux résoudre le ou les problèmes.

L'architecture Model-Vu-Contrôleur

L'architecture MVC (modèle, vue et contrôleur) est un concept très puissant qui intervient dans la réalisation d'une application ou un site web. Son principal intérêt est la séparation des données (modèle), de l'affichage (vue) et des actions (contrôleur).



Lorsqu'une requête arrive au serveur de Laravel, qu'elle s'agit d'une demande de page web, le fichier routes.php charge le contrôleur nécessaire qui correspond à l'URL, ainsi il charge toutes les informations nécessaires pour fournir une réponse à l'utilisateur. La vue reçoit les données depuis le contrôleur et génère la structure HTML pour l'utilisateur, elle charge également le code CSS et Javascript pour cette page.

1. Le Modèle : Représente le comportement de l'application : traitements des données, interactions avec la base de données, etc. Il décrit les données manipulées par l'application et définit les méthodes d'accès.
2. La Vue : Correspond à l'interface avec laquelle l'utilisateur interagit. Les résultats renvoyés par le modèle sont dénués de toute présentation mais sont présentés par les vues. Plusieurs vues peuvent afficher les informations d'un même modèle. Elle peut être conçue en html, ou tout autre " langage " de présentation. La vue n'effectue aucun traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle, et de permettre à l'utilisateur d'interagir avec elles.
3. Le Contrôleur : Prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle. Il n'effectue aucun traitement, ne modifie aucune donnée, il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondante à la demande.

Extraction du moteur d'analyse

Nous avons procédé par élimination pour extraire le noyau du projet qu'est l'analyseur, ce procédé consiste à éliminer les éléments du routage, ainsi que les fonctions dédiées à l'authentification et l'enregistrement, ainsi que le style du projet pour ensuite finir avec les fonctions dédiées à l'analyseur.

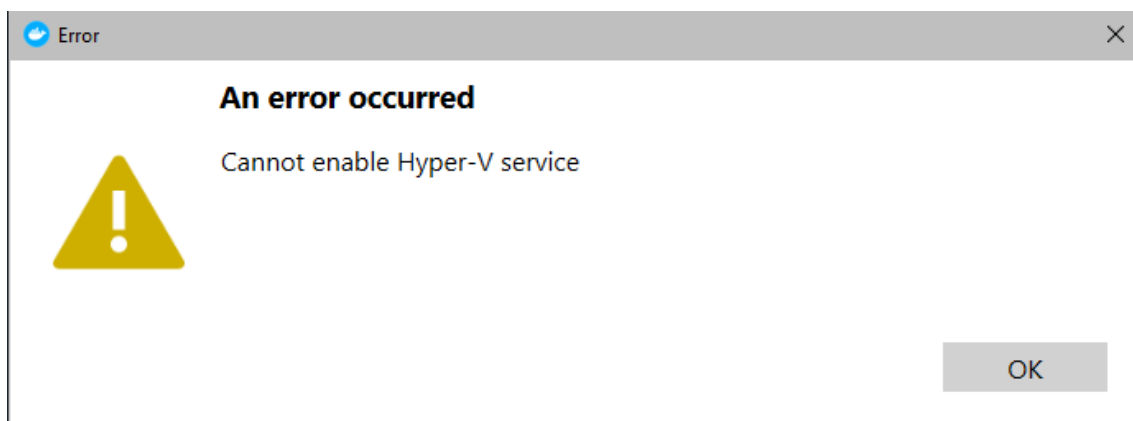
Le Docker

Est un outil qui peut emballer une application et ses dépendances dans un conteneur virtuel, qui pourra être exécuté sur n'importe quel serveur Linux



Installation

L'installation du Docker sur Windows 10 se fait par l'intermédiaire d'un exécutable. Il est possible de le télécharger sur le site officiel du logiciel à l'adresse <https://www.docker.com/get-started>, l'installation se fait de la même manière avec le système Linux en utilisant la ligne de commandes. Il suffit ensuite de l'exécuter en mode administrateur et de suivre les instructions. Docker Desktop a l'avantage de fournir une interface graphique simplifiant l'utilisation des conteneurs. Lors de l'utilisation de Docker sur une version de Windows, vous pourriez rencontrer une erreur comme celle sur la capture ci-dessous.



Hyper-V est un service de virtualisation utilisé par Windows et l'erreur provient du fait que le programme n'y a pas accès. Pour modifier cela, il faut suivre les étapes suivantes :

1. Ouvrir la barre de recherche à l'aide du raccourci "Touche Windows + R" et rechercher "optionalfeatures".
2. Cocher toutes fonctionnalités mentionnant "Hyper-V", "Virtual Machine Platform" et "Windows Hypervisor Platform". Toutes ces fonctionnalités ne sont pas toujours disponibles selon la version de votre système d'exploitation.
3. Redémarrer l'ordinateur.

Docker nécessite également d'installer le "Windows Subsystem for Linux" qui est littéralement un sous-système Linux pour Windows. Cette installation passe par cinq étapes :

1. Ouvrez l'invite de commande Powershell en mode administrateur et tapez : `dism.exe /online /enable-feature /featurename :Microsoft-Windows-Subsystem-Linux /all /norestart`

2. Toujours dans Powershell, entrez : `dism.exe /online /enable-feature /featurename :VirtualMachinePlatform /all /norestart` Avant de redémarrer votre ordinateur
3. Télécharger et installez le paquet mis à disposition à l'adresse
4. A l'aide de Powershell, exécutez la commande `wsl -set-default-version 2`.
5. Ouvrez le Microsoft Store et installez la distribution de linux qui vous intéresse, par exemple Ubuntu 20.04LTS.

Si un problème se déroule lors de cette installation, nous vous conseillons de consulter la documentation officielle disponible à l'adresse <https://docs.microsoft.com/en-us/windows/wsl/install-win10>.

Implémentation

Implémentation du fichier `docker-compose.yml`

L'intégration du docker se fait avec 3 micro-services : Ces trois dernier sont mis en place grace a un fichier yml créé a la racine du projet. On y trouve :

1. Un micro service « app » : on y retrouve plusieurs paramètres :
 - `build` : cette configuration indique à Docker Compose d'utiliser le chemin spécifié (contexte) et Dockerfile comme instructions pour créer une image locale pour le service d'application. La personnalisation des accès est assurée par la définition de l'uid et le user.
 - `image` : le nom qui sera utilisé pour l'image en cours de construction.
 - `containername` : définit le nom du conteneur pour ce service.
 - `restart` : indique qu'il faut toujours redémarrer, sauf si le service est arrêté.
 - `workingdir` : définit le répertoire par défaut pour ce service comme `/var/www`.
 - `volumes` : crée un volume partagé qui synchronisera le contenu du répertoire actuel avec `/var/www` dans le conteneur. Notez qu'il ne s'agit pas de la racine de votre document, puisqu'elle se trouve dans le conteneur nginx.
 - `networks` : configure ce service pour utiliser un réseau nommé `analyseweb`.
1. Un micro service « mysql » Puisque Docker Compose charge automatiquement le fichier variable `.env` dans le même répertoire que le fichier `docker-compose.yml`, nous pouvons obtenir les paramètres de la base de données à partir du fichier `.env` Laravel mis en place pour connecter une base de données au projet selon la logique laravel. Ce micro-service est défini ainsi :
 - `image` : définit l'image Docker qui doit être utilisée pour ce conteneur.
 - `containername` : définit le nom du conteneur
 - `restart` : indique qu'il faut toujours redémarrer ce service, sauf s'il est explicitement arrêté.
 - `Environnement` : définissez les variables d'environnement dans le nouveau conteneur. Nous utilisons les valeurs obtenues à partir du fichier `.env` pour configurer notre service MySQL, qui créera automatiquement une nouvelle base de données et un nouvel utilisateur en fonction des variables d'environnement fournies avec des une structure similaire a la table `analyseweb,analysis`.
 - `volumes` : crée un volume pour un système de partage de données qui seront utilisées pour initialiser la base de données de l'application.
 - `networks` : configure ce service pour utiliser un réseau nommé `analyseweb`.
1. - Un micro service « nginx » Le service Nginx utilise une image Nginx pré-construite sur Alpine (une distribution Linux légère). Il crée un conteneur nommé `analyseweb-nginx` et

utilise la définition de port pour créer une redirection du port hôte 8000 vers le port 80 sur le système hôte.

Implémentation et la configuration Nginx

L'utilisation du Docker Compose pour créer un environnement de développement, requière généralement le partage des fichiers de configuration ou l'initialisation avec le conteneur de services afin de configurer ou de démarrer ces services. De cette manière, les fichiers de configuration peuvent être facilement modifiés pour affiner l'environnement lors du développement d'applications. d'où la creation d'un dossier contenant des fichiers qui seront utilisés pour configurer et initialiser les conteneur de services. Pour configurer Nginx, un fichier analyseweb.conf est partagé afin de configurer la façon dont l'application est servie, Ce fichier configurera Nginx pour écouter sur le port de lancemant et utilisera index.php comme page d'index par défaut. Il définira le document root comme /var/www/public, et configurera ensuite Nginx pour utiliser le service de l'application sur un port pré-défini pour traiter les fichiers *.php. Un dossier « mysql » est crée dans le même dossier, ce dernier contient un fichier .sql avec une définition d'une table parfaitement identique a la table analyseweb,analysis, qui récoltera les données resultant de chaque analyse.

La mise en place du Dockerfile

Bien que les images utilisé pour les service nginx et mysql sont des image standards du docker hub, il reste nécessaire de créer une image personnalisée, Le Dockerfile est créé a la racine du projet,

- on commence par définir une image mère, l'image dont notre image personnalisée sera héritière,
- puis communiquer l'iud et le user pré défini dans le docker-compose.yml
- en suite on doit installer les dépendances système liées au projet. Dans notre cas surtout les dépendance php comme curl afin d'assurer le fonctionnement de laravel qu'est un framework a noyau php .
- en suite on doit vider le cash afin d'éviter tout problèmes lié a la gérance d'espace ou même ralentissement.
- puis l'installation des extensions php a utiliser.
- ensuite recuperer la version la plus recente du composer afin de ne pas créer des conflit de version ou package avec le projet d'origine.
- configurations lié a l'utilisation des commandes lié a laravel telle que « composer » et « artisan »
- définition du répertoire de travail de réceptionner le build final de la mise en place de la virtualisation.

La construction

La construction se fait en deux étapes :

- Cette première étape consiste à construire la structure des micro services, et les relier au projet laravel.


```

lydia@lydia-Inspiron-15-3567:~/univ/Analyse web/analyse-web$ sudo docker-compose build app
[sudo] password for lydia:
Building app
Step 1/11 : FROM php:7.4-fpm
--> 11d1c5c37e37
Step 2/11 : ARG user
--> Using cache
--> c5e08ae5dc46
Step 3/11 : ARG uid
--> Using cache
--> de9bfe121194
Step 4/11 : RUN apt-get update && apt-get install -y git curl libpng-dev libonig-dev libxml2-dev zip unzip
--> Using cache
--> 3d700ddb844d
Step 5/11 : RUN apt-get clean && rm -rf /var/lib/apt/lists/*
--> Using cache
--> 88cd05b42680
Step 6/11 : RUN docker-php-ext-install pdo_mysql mbstring exif pcntl bcmath gd
--> Using cache
--> ec8698b84d69
Step 7/11 : COPY --from=composer:latest /usr/bin/composer /usr/bin/composer
--> Using cache
--> 216f9b6944fb
Step 8/11 : RUN useradd -G www-data,root -u $uid -d /home/$user $user
--> Using cache
--> 2624d3b0aa16
Step 9/11 : RUN mkdir -p /home/$user/.composer && chown -R $user:$user /home/$user
--> Using cache
--> 407fee89bd27
Step 10/11 : WORKDIR /var/www
--> Using cache
--> c4d91e128f51
Step 11/11 : USER $user
--> Using cache
--> 1c58cf3b8602
Successfully built 1c58cf3b8602
Successfully tagged analyseweb:latest

```

- La deuxième étape consiste à mettre en place l’environnement et installer les dépendances nécessaires pour la virtualisations (les paquages et les commandes d’exécution).

```

lydia@lydia-Inspiron-15-3567:~/univ/Analyse web/analyse-web$ sudo docker-compose up -d
analyseweb-db is up-to-date
analyseweb-nginx is up-to-date
analyseweb-app is up-to-date

```

Les avantages

- ça rend la technologie beaucoup plus portable
- c’est facilement scalable c’est à dire on va pouvoir bouger les petits blog d’un endroit à un autre et les faire évoluer en terme de performances
- meilleures performances
- Le déploiement est très simple vu qu’un contener est légée donc l’envoyer par exemple à un hebergeur il va juste le faire fonctionner.

Cette technologie est de plus en plus interessante grace à docker car il permet d’utiliser le système de contener d’une manière simple il propose une série d’outils à utiliser pour ne pas se soucier du fonctionnement, déploiement ...le docker s’en charge de tout.

La différence entre Le Docker et La VM

La prise en charge du système d’exploitation de la machine virtuelle et du conteneur Docker est très différente. La machine virtuelle a son système d’exploitation invité au-dessus du système d’exploitation hôte, ce qui alourdit les machines virtuelles. Alors que d’un autre côté, les conteneurs Docker partagent le système d’exploitation hôte, et c’est pourquoi ils sont légers. Le partage du système d’exploitation hôte entre les conteneurs les rend très légers et les aide à

démarrer en quelques secondes. Par conséquent, la surcharge de gestion du système de conteneurs est très faible par rapport à celle des machines virtuelles.

La Virtualisation

La virtualisation consiste, en informatique, à exécuter sur une machine hôte, dans un environnement isolé, des systèmes d'exploitation. La virtualisation des données est une approche permettant d'unifier les données de plusieurs sources dans une même couche afin que les applications, les outils de génération de rapports et les utilisateurs finaux puissent accéder aux données sans avoir besoin de détails sur la source, l'emplacement et les structures de données d'origine.

Conclusion

Ce projet nous a permis de nous familiariser avec le framework Laravel ainsi qu'en apprendre d'avantages sur la technologie des dockers qui aujourd'hui a une grande importance dans le monde de l'informatique dû à ses capacités de traiter les données de façon à réduire le coût et le temps d'exécution avec une structure plus au moins légère et modulaire. Nous avons pu mettre en pratique nos cours de système d'exploitation et de PHP et de BDD. Bien que le projet soit encore améliorable, nous avons réussi à mettre en place une base solide pour une exploitation du docker en extractant le moteur d'analyse et en mettant en place les outils nécessaires pour finaliser la virtualisation. Ça nous a permis certainement de développer et de mettre au défi nos compétences interpersonnelles pour bien mener ce projet avec un groupe aussi nombreux en se basant sur une organisation arrangeante pour tout le monde en utilisant des outils informatique comme discord, messenger et les e-mails.

Bibliographie

- <https://docs.docker.com/get-started/>
- <https://docs.docker.com/get-docker/>
- <https://laravel.sillo.org/>
- <https://laravel.com/docs/8.x/structure>
- <https://doc.ubuntu-fr.org/docker>
- <https://laracasts.com/series/laravel-8-from-scratch/episodes/1>
- <https://docs.docker.com/get-started/>
- <https://www.youtube.com/watch?v=XgKOC6X8W28&t=496s>
- <https://stackoverflow.com/>
- <https://www.digitalocean.com/community/tutorials/how-to-set-up-laravel-nginx-and-mysql-with-docker-compose>
- <https://openclassrooms.com/>
- <https://www.youtube.com/watch?v=8kXz76QpGj8&t=914s>
- <https://www.youtube.com/watch?v=xw1rmQDeeV8&t=109s>
- <https://www.youtube.com/watch?v=-NYKvQqI1TA>